Course Overview

- Introduction and History
- Data in Wireless Cellular Systems
- Data in Wireless Local Area Networks
- Internet Protocols
- Routing and Ad-Hoc Networks
- TCP over Wireless Link
- Services and Service Discovery
- System Support for Mobile Applications

What Exactly does "Mobile" mean?

- user mobility vs. computer mobility
  - user mobility: always work in same computing environment, problem is how to "teleport" environment to location of user
  - computer mobility: computing environment moves with user, utilizes wireless links, has to deal with changing environments
- scales of mobility:
  - within house/building
  - campus/town/city
  - state or region
  - country or continent
  - world
- we will look at potentially world-wide computer mobility

Challenges in Mobile Computing

- wireless communication is substantially different from wired communication:
  - frequent spurious disconnections (handoff, shadowed areas)
  - lower bandwidth (GSM, CDPD: less than 20 kbps)
  - high bandwidth variability (roaming outdoors vs. plugged into docking station)
  - heterogeneous networks (outdoors: cellular, indoors: infrared LAN)
  - security risks (everyone can listen in on wireless channel)
- mobility:
  - address migration (e.g., location management)
  - location dependent information (where is closest printer?)
  - migrating locality (optimal connection changes over time)

Challenges in Mobile Computing

- portability (of computing device):
  - limited power supply (battery)
  - data security (someone steals your laptop)
  - small user interface (look at screen size of a PDA)
  - limited storage capacity (disks, for example, are a “power liability”)
- solutions (or suggestions) exist to address all these issues, however, the solutions are often conflicting:
  - do not keep data on portable device to reduce security problem
  - do not send data if not necessary to save valuable energy

Solutions and Research Directions

- solutions to overcome hardware disparities:
  - sophisticated power management (sleep modes, disk mgmt, clock speed, prefer receiving data over sending: “ether as cache”)
  - delegate tasks to stationary computers (application partitioning, stationary computer has disk space, high bandwidth connection, powerful CPU)
  - trade increased data processing for reduced network bandwidth requirements (on-line compression, difference-based updates, filtering, less restrictive consistency semantics)
  - reduce network latency (data prefetching, batch multiple messages)

Solutions and Research Directions

- solutions addressing mobility:
  - flexible directory services to enable mobile to find location-dependent services, often based on “shopping” or negotiation:
    - scaleable, fault-tolerant architecture (exploit communication locality, replicate directory data, weakened consistency semantics: allow directory info to go stale as long as we can improve on suboptimal routes during connection)
    - brokerage service (e.g.: trading in ODP)
    - accounting service: mediate cost accounting between mobile client and servers
    - rebindable service interfaces (switch service providers “on-line”)
  - ubiquitous security and authentication (IPv6 will go a long way towards providing this goal)

Solutions and Research Directions

- there seems to be general agreement that some sort of application partitioning is a good thing:
  - reduce bandwidth requirements
  - overcome the physical limitations of the mobile computer
- one open question: how do you partition and when?
  - static partitioning is easier (and we will see examples for WWW later)
  - however, computation environment can change radically during execution:
    - MH moves to area with better networking infrastructure (maybe gets back into his docking station)
    - PCMCIA cards all of a sudden add more memory
    - radio signal is dropped completely temporarily
  - implies growing interest in dynamic application partitioning

A Formal Model

- one way to examine issues in designing algorithms for a mobile environment is to develop and use a formal model (Badrinath et al.):
  - network consists of MH and MSS (Mobile Support Stations)
  - MSS covers a cell, uses broadcast within cell
  - MSS connected via a wired backbone
  - to communicate between two mobile hosts MH1 and MH2, message is routed through the corresponding MSSs
  - MSS may have to determine location of serving MSS for outgoing call through broadcast
  - MH are mobile, no time bound between leaving a cell and entering the next
  - FIFO MH-MSS channel within cell. MH typically receives only prefix of the messages sent (might have left cell)

A Formal Model

- some consequences from model:
  - communication costs are no longer uniform:
    - relatively cheap to communicate between two MSS
    - relatively expensive to communicate MSS-MH or MH1-MH2
  - mobile hosts may disconnect (or doze off), reappear later
  - underlying network structure changes all the time
- model applied to study a well-known distributed mutual exclusion algorithm (Lamport's Algorithm):
  - if original algorithm implemented directly, poor performance:
    - high search costs for all MH-MH messages
    - all hosts required to participate, so disconnections not allowed
    - data structures need to be maintained at the mobile hosts, so high power usage

A Formal Model

- redesign of Lamport's Mutual Exclusion Algorithm:
  - two-tier structure: most of the coordination between MSS only
  - if MH wants to enter critical section, it contacts local MSS
  - local MSS obtains permission to enter critical section, which it passes on to requesting mobile
  - if requesting mobile leaves cell, update status during handoff (the model provides for "leave" messages when stations disconnect)
  - result: most communication limited to fast backbone, mobile stations are allowed to disconnect and to doze off
- using two-tier strategy is basically similar to "application partitioning"

Typical Applications

- you most likely will **not** use your PDA to solve the *grand challenge* problems (after all, this is what MPPs and supercomputers are there for)
- unifying theme: mobile computer is interface to information anywhere at any time:
  - "access ramp" to NII
  - AOAC (always on, always connected)
- at least four main classes of applications
  - information browsing
  - personal communication
  - multiperson interaction
  - data entry

Typical Applications

- information browsing:
  - querying traditional databases
  - retrieving information from electronic books, magazines, newspapers (Newsnet News)
  - navigating through hypertext and other interactive documents (Gopher, Wais, WWW)
- personal communication:
  - e-mail and fax
  - key: from any location, at any time: asynchronously
- multiperson interaction:
  - groupware applications, electronic whiteboard (Bell commercial)
- data entry:
  - on-site entry: car rental agencies, warehouses, archaeology

File Systems: Motivation

- Goal
  - efficient and transparent access to shared files within a mobile environment while maintaining data consistency
- Problems
  - limited resources of mobile computers (memory, CPU, ...)
  - low bandwidth, variable bandwidth, temporary disconnection
  - high heterogeneity of hardware and software components (no standard PC architecture)
  - wireless network resources and mobile computer are not very reliable
  - standard file systems (e.g., NFS, network file system) are very inefficient, almost unusable
- Solutions
  - replication of data (copying, cloning, caching)
  - data collection in advance (hoarding, pre-fetching)

File Systems: Consistency Problems

- THE big problem of distributed, loosely coupled systems
  - are all views on data the same?
  - how and when should changes be propagated to what users?
- Weak consistency
  - many algorithms offering strong consistency (e.g., via atomic updates) cannot be used in mobile environments
  - invalidation of data located in caches through a server is very problematic if the mobile computer is currently not connected to the network
  - occasional inconsistencies have to be tolerated, but conflict resolution strategies must be applied afterwards to reach consistency again
- Conflict detection
  - content independent: version numbering, time-stamps
  - content dependent: dependency graphs

File Systems: Limited Connectivity

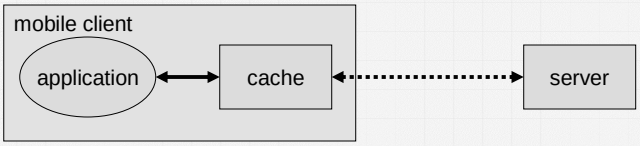
- Symmetry
  - Client/Server or Peer-to-Peer relations
  - support in the fixed network and/or mobile computers
  - one file system or several file systems
  - one namespace for files or several namespaces
- Transparency
  - hide the mobility support, applications on mobile computers should not notice the mobility
  - user should not notice additional mechanisms needed
- Consistency model
  - optimistic or pessimistic
- Caching and Pre-fetching
  - single files, directories, subtrees, partitions, ...
  - permanent or only at certain points in time

File Systems: Limited Connectivity

- Data management
  - management of buffered data and copies of data
  - request for updates, validity of data
  - detection of changes in data
- Conflict solving
  - application specific or general
  - errors
- Several experimental systems exist
  - Coda (Carnegie Mellon University), Little Work (University of Michigan), Ficus (UCLA) etc.
- Many systems use ideas from distributed file systems such as, e.g., AFS (Andrew File System)

File Systems: Coda

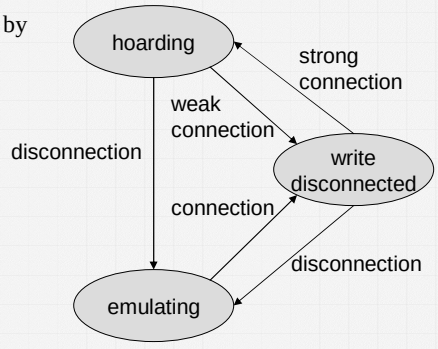
- Application transparent extensions of client and server
  - changes in the cache manager of a client
  - applications use cache replicates of files
  - extensive, transparent collection of data in advance for possible future use („Hoarding“)
- Consistency
  - system keeps a record of changes in files and compares files after reconnection
  - if different users have changed the same file a manual reintegration of the file into the system is necessary
  - optimistic approach, coarse grained (file size)



File Systems: Coda

- Hoarding
  - user can pre-determine a file list with priorities
  - contents of the cache determined by the list and LRU strategy (Last Recently Used)
  - explicit pre-fetching possible
  - periodic updating
- Comparison of files
  - asynchronous, background
  - system weighs speed of updating against minimization of network traffic
- Cache misses
  - modeling of user patience: how long can a user wait for data without an error message?
  - function of file size and bandwidth

- States of a client



File Systems: Other Examples

- Mazer/Tardo
  - file synchronization layer between application and local file system
  - caching of complete subdirectories from the server
  - "Redirector" responses to requests locally if necessary, via the network if possible
  - periodic consistency checks with bi-directional updating
- Ficus
  - not a client/server approach
  - optimistic approach based on replicates, detection of write conflicts, conflict resolution
  - use of „gossip" protocols: a mobile computer does not necessarily need to have direct connection to a server, with the help of other mobile computers updates can be propagated through the network
- MIo-NFS (Mobile Integration of NFS)
  - NFS extension, pessimistic approach, only token holder can write
  - connected/loosely connected/disconnected

Database Systems in Mobile Environments

- Request processing
  - power conserving, location dependent, cost efficient
  - example: find the fastest way to a hospital
- Replication management
  - similar to file systems
- Location management
  - tracking of mobile users to provide replicated or location dependent data in time at the right place (minimize access delays)
  - example: with the help of the HLR (Home Location Register) in GSM a mobile user can find a local towing service
- Transaction processing
  - "mobile" transactions can not necessarily rely on the same models as transactions over fixed networks (ACID: atomicity, consistency, isolation, durability)
  - therefore models for "weak" transaction

World Wide Web and Mobility

- Protocol (HTTP, Hypertext Transfer Protocol) and language (HTML, Hypertext Markup Language) of the Web have not been designed for mobile applications and mobile devices, thus creating many problems!
- Typical transfer sizes
  - HTTP request: 100-350 byte
  - responses avg. <10 kbyte, header 160 byte, GIF 4.1kByte, JPEG 12.8 kbyte, HTML 5.6 kbyte
  - but also many large files that cannot be ignored
- The Web is no file system
  - Web pages are not simple files to download
  - static and dynamic content, interaction with servers via forms, content transformation, push technologies etc.
  - many hyperlinks, automatic loading and reloading, redirecting
  - a single click might have big consequences!

HTTP 1.0 and Mobility

- Characteristics
  - stateless, client/server, request/response
  - needs a connection oriented protocol (TCP), one connection per request (some enhancements in HTTP 1.1)
  - primitive caching and security
- Problems
  - designed for large bandwidth (compared to wireless access) and low delay
  - big and redundant protocol headers (readable for humans, stateless, therefore big headers in ASCII)
  - uncompressed content transfer
  - using TCP
    - huge overhead per request (3-way-handshake) compared with the content, e.g., of a GET request
    - slow-start problematic
  - DNS lookup by client causes additional traffic

HTTP 1.0 and Mobility

- Caching
  - quite often disabled by information providers to be able to create user profiles, usage statistics etc.
  - dynamic objects cannot be cached
    - numerous counters, time, date, personalization, ...
  - mobility quite often inhibits caches
  - security problems
    - how to use SSL/TLS together with proxies?
  - today: many user customized pages, dynamically generated on request via CGI, ASP, ...
- POSTing (i.e., sending to a server)
  - can typically not be buffered, very problematic if currently disconnected
- Many unsolved problems!

HTML and Mobile Devices

- HTML
  - designed for computers with “high” performance, color high-resolution display, mouse, hard disk
  - typically, web pages optimized for design, not for communication
- Mobile devices
  - often only small, low-resolution displays, very limited input interfaces (small touch-pads, soft-keyboards)
- Additional “features”
  - animated GIF, Java AWT, Frames, ActiveX Controls, Shockwave, movie clips, audio, ...
  - many web pages assume true color, multimedia support, high-resolution and many plug-ins
- Web pages ignore the heterogeneity of end-systems!
  - e.g., without additional mechanisms, large high-resolution pictures would be transferred to a mobile phone with a low-resolution display causing high costs

WWW for Mobile Devices

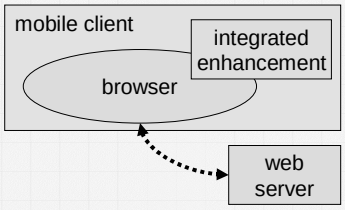
- Application gateways, enhanced servers
  - simple clients, pre-calculations in the fixed network
  - compression, filtering, content extraction
  - automatic adaptation to network characteristics
- Examples
  - picture scaling, color reduction, transformation of the document format (e.g., PS to TXT)
  - detail studies, clipping, zoom
  - headline extraction, automatic abstract generation
  - HDML (handheld device markup language): simple language similar to HTML requiring a special browser
  - HDTP (handheld device transport protocol): transport protocol for HDML, developed by Unwired Planet
- Problems
  - proprietary approaches, require special enhancements for browsers
  - heterogeneous devices make approaches more complicated

New Issues that might help Mobility?

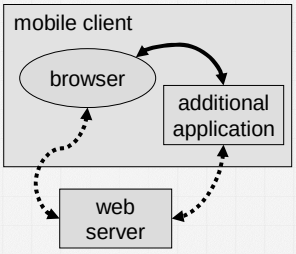
- Push technology
  - real pushing, not a client pull needed, channels etc.
- HTTP/1.1
  - client/server use the same connection for several request/response transactions
  - multiple requests at beginning of session, several responses in same order
  - enhanced caching of responses (useful if equivalent responses!)
  - semantic transparency not always achievable: disconnected, performance, availability -> most up-to-date version...
  - several more tags and options for controlling caching (public/private, max-age, no-cache etc.)
  - relaxing of transparency on app. request or with warning to user
  - encoding/compression mechanism, integrity check, security of proxies, authentication, authorization...
- Cookies: well..., stateful sessions, not really integrated...

System Support for Mobile WWW

- Enhanced browsers
  - Pre-fetching, caching, off-line use
  - e.g. Internet Explorer

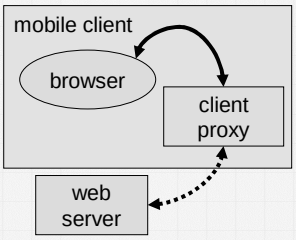


- Additional, accompanying application
  - Pre-fetching, caching, off-line use
  - e.g. original WebWhacker

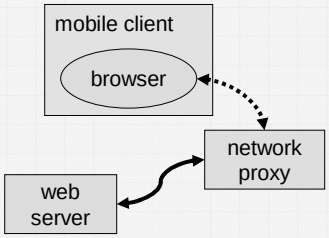


System Support for Mobile WWW

- Client Proxy
  - Pre-fetching, caching, off-line use
  - e.g., Caubweb, TeleWeb, Weblicator, WebWhacker, WebEx, WebMirror, ...

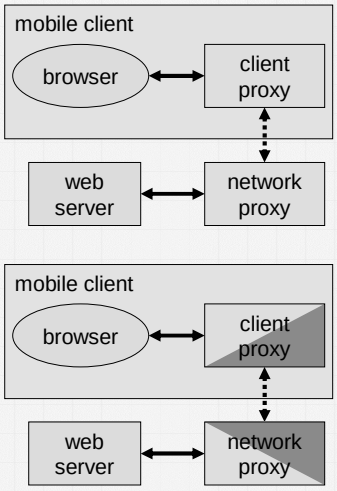


- Network Proxy
  - adaptive content transformation for bad connections, pre-fetching, caching
  - e.g., TranSend, Digestor



System Support for Mobile WWW

- Client and network proxy
  - combination of benefits plus simplified protocols
  - e.g., MobiScape, WebExpress
- Special network subsystem
  - adaptive content transformation for bad connections, pre-fetching, caching
  - e.g., Mowgli
- Additional many proprietary server extensions possible
  - “channels”, content negotiation, ...



WAP Forum

(see also `http://www.wapforum.org/`)

- “de-facto world standard for wireless information and telephony services on digital mobile phones and other wireless terminals”
- published a global wireless protocol specification based on existing Internet standards such as XML and IP for all wireless network
- Marketing Cloud:
  - Handset manufacturers representing over 75% of the world market across all technologies have committed to shipping WAP-enabled devices.
  - Carriers representing nearly 100 million subscribers worldwide have joined WAP Forum

WAP Forum: History

- Started in June 1997
- Founders: phone.com (Unwired Planet), Ericsson, Nokia, Motorola
- Current membership: 150+ companies
  - Carriers, hardware, software, content developers
  - For example: HP, IBM, NEC, Lucent, Nortel Networks, Sun, Siemens

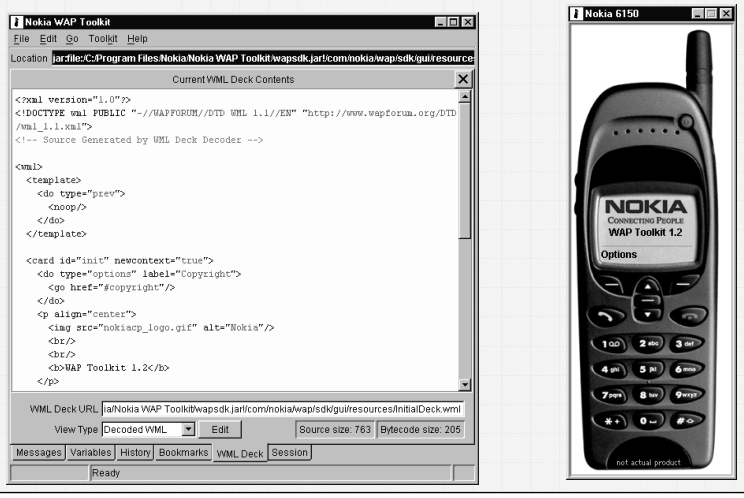
WAP Specifications

- Current devices are Version 1.0
- Current servers and services are 1.0 or 1.1
- Version 1.1 recently approved (which is version we will be discussing)
- Specifications very ambitious/all-encompassing:
  - promote industry-wide standards and specifications for developing applications and services that operate over wireless communication networks
  - aims to enable operators, manufacturers, and content developers "to meet the challenges of implementing advanced differentiating services and applications in a fast and flexible manner"

WAP Forum Goals and Assumptions

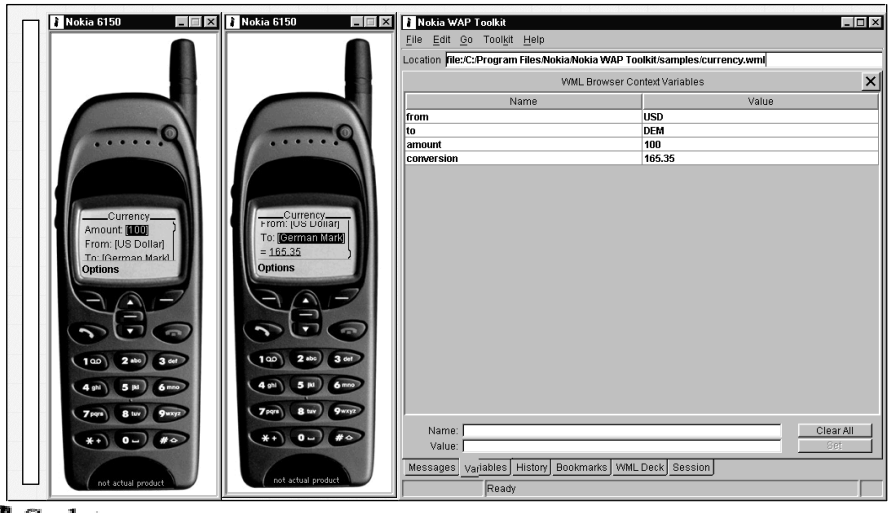
- Goals:
  - Provide Internet access to consumer-class wireless devices
  - Create a global protocol specification that works on all wireless networks
  - Current focus is on cell phones, the most widely available consumer wireless device
- Assumptions:
  - Limited processing capabilities, RAM, ROM, battery life
  - Small screens and limited data entry capabilities
  - Low-power, modest bandwidth wireless networks
  - Networks inherently unstable, unreliable, and unpredictable

WAP Example: Nokia SDK



WAP Example: Weather Forecast

WAP Example: Currency Conversion



WAP Example: Different Handsets

Same application:
Handset is free to render data and control user interaction in different ways

WAP Example: Source Code (WML)

```
<wml>
<card id="card1" title="Currency" newcontext="true">
        <p>
        Amount: <input format="*N" name="amount" title="Amount:"/>
        From: <select name="from" value="USD" title="From:">
                <option value="DEM">German Mark</option>
                <option value="FRF">French Franc</option>
                <option value="FIM">Finnish Markka</option>
                <option value="USD">US Dollar</option>
        </select>
        To: <select name="to" value="FIM" title="To:">
                <option value="DEM">German Mark</option>
                <option value="FRF">French Franc</option>
                <option value="FIM">Finnish Markka</option>
                <option value="USD">US Dollar</option>
        </select>
        <br/> = <u>$(conversion)</u>
        <do type="accept" label="Calculate">
                <go href="currency.wmls#convert('conversion','$(from)','$(to)',$(amount))"/>
        </do>
        <do type="help" label="Help">
                <go href="#card1_help"/>
        </do>
        </p>
</card>
```

WAP Example: Source Code (cont.)

```
<card id="card1_help" title="Help">
        <onevent type="onenterforward">
                <go href="currency.wmls#getInfoDate('date')"/>
        </onevent>
        <p>
                The currency rates were obtained from the Federal
                Reserve Bank of New York on $(date).
                <do type="prev" label="Back">
                        <prev/>
                </do>
        </p>
</card>
</wml>
```


WAP Suite

Wireless Application Environment Specification [WAE]:
The Wireless Application Environment specification is the root document in the WAE normative document hierarchy. The document specifies and references core WAE elements.

Wireless Markup Language Specification [WML]
The Wireless Markup Language specification describes the mark-up language, WML, including its semantics, its document type definition (DTD) and its encoding extensions.

WAP Binary XML Format Specification [WBXML]:
The WAP Binary XML Forum specification describes the XML document encoding and transfer framework used by WAE.

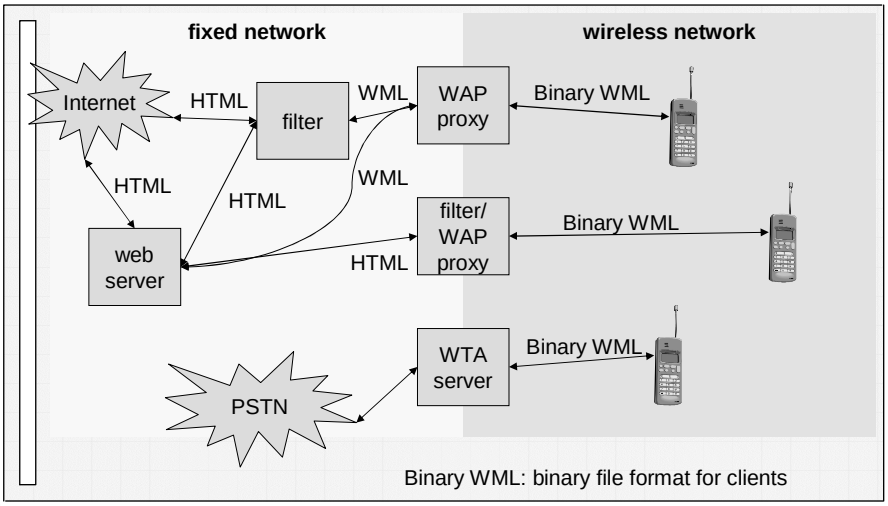
WMLScript Specification [WMLScript]:
The WMLScript specification describes the scripting language, WMLScript, including its lexical and syntactic grammar, its transfer format and a reference bytecode interpreter.

WMLScript Standard Libraries Specification [WAEStdLib]:
The WMLScript Standard Libraries specification describes standard libraries available to WMLScript programs including a language library, a string library, a dialog library, a floating-point library, a browser library and a URL library.

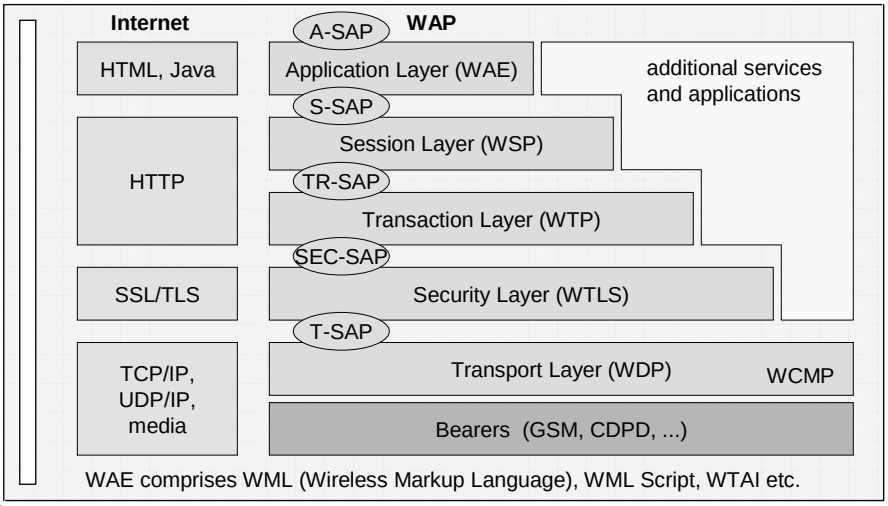
Wireless Telephony Application Specification [WTA]:
The Wireless Telephony Application Specification specifies the technologies included in the Wireless Telephony Application reference architecture.

Wireless Telephony Application Interface [WTAI]
The Wireless Telephony Application Interface specification describes standard telephony-specific extensions to WAE including WML and WMLScript interfaces to such items as call control features, address book and phonebook services.

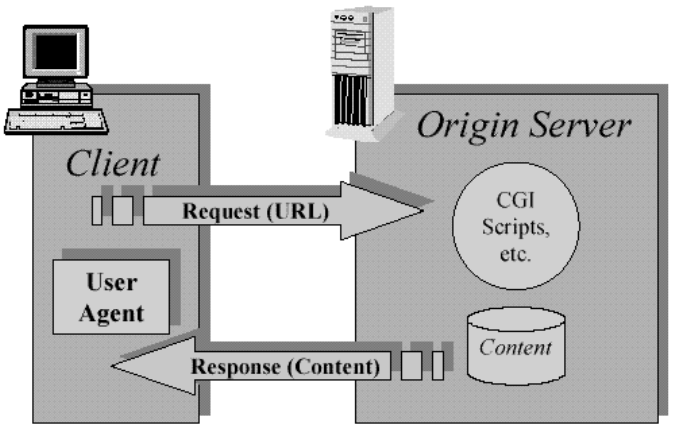
WAP Network Elements



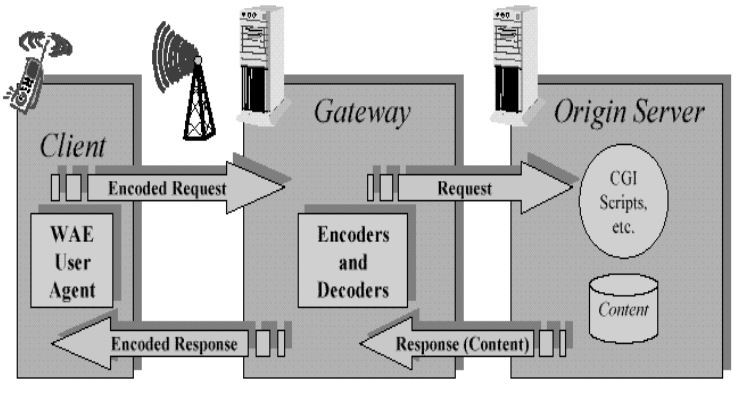
WAP Reference Model and Protocols



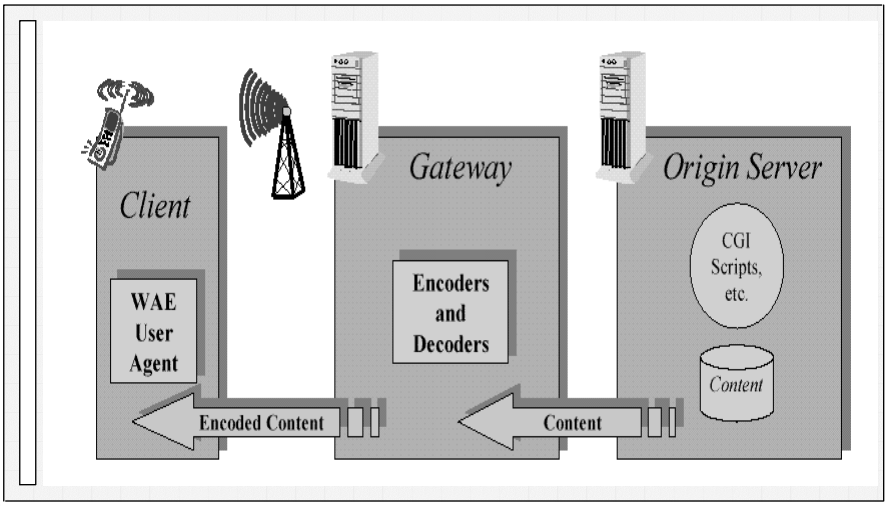
WAE Architecture Overview: The WWW Logical Model



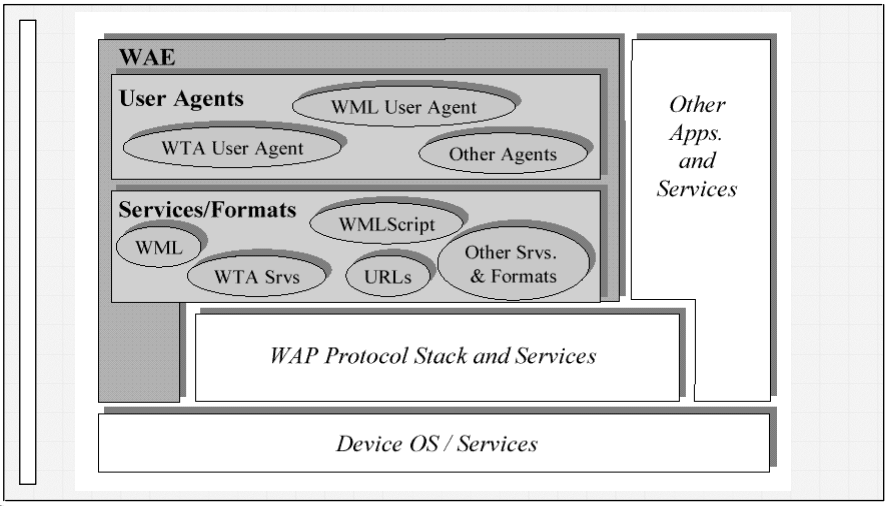
WAE Architecture Overview: The WAE Logical Model



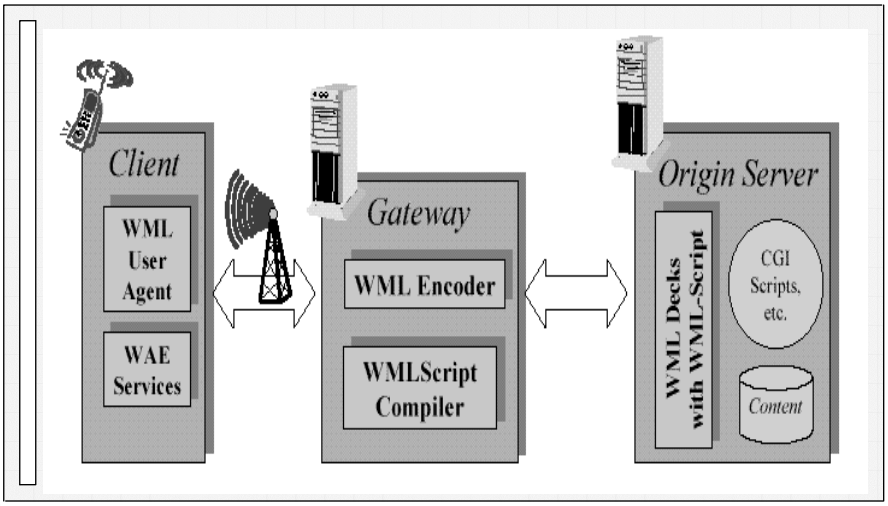
WAE Architecture Overview: The WAE Push-based Model



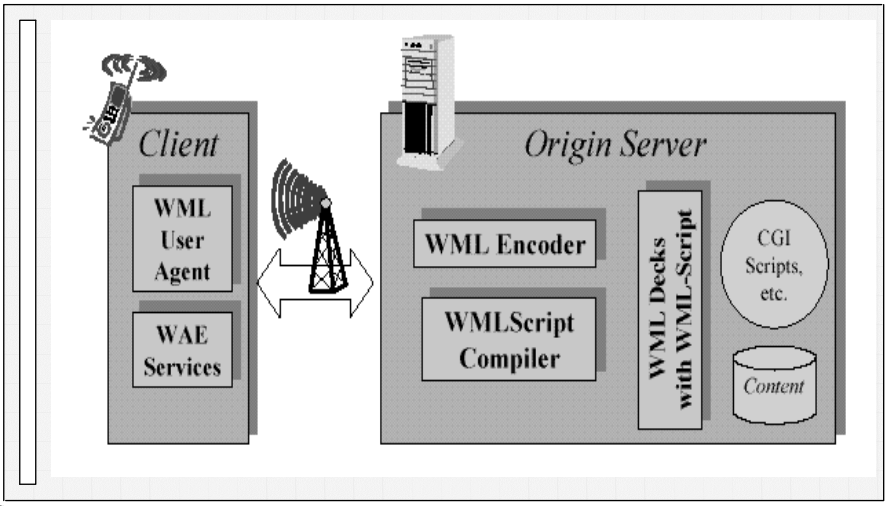
WAE Client Components



WML User Agent Logical Architecture



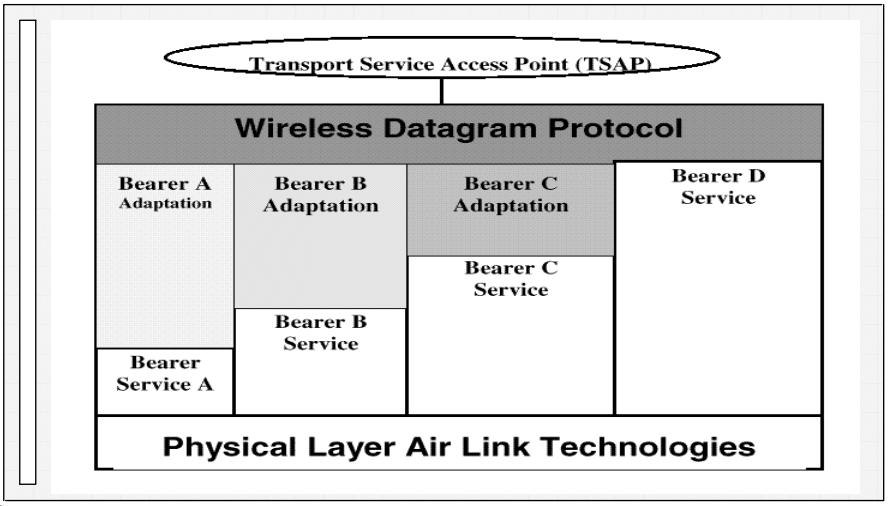
WML User Agent Logical Architecture without a Gateway



WAP Protocol Stack

	WTP User (e.g. WSP) — WSP: Wireless Session Protocol
WTP Wireless Transaction Protocol	❑ Transaction handling ❑ Re-transmissions, duplicate removal, acknowledgements ❑ Concatenation and separation
[WTLS] Wireless Transport Layer Security	❑ Optionally compression ❑ Optionally encryption ❑ Optionally authentication
Datagram Transport (e.g. WDP) Wireless Datagram Protocol	❑ Port number addressing ❑ Segmentation and re-assembly (if provided) ❑ Error detection (if provided)
Bearer Network (e.g. IP, GSM SMS/USSD, IS-136 GUTS)	❑ Routing ❑ Device addressing (IP address, MSISDN) ❑ Segmentation and re-assembly (if provided) ❑ Error detection (if provided)

Wireless Datagram Protocol (WDP)



WDP Architecture